

FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

The following is a complete listing of all claims in the application, with an indication of the status of each:

**Listing of claims:**

1                   1. (currently amended) ~~A~~ An automated method for generating  
2                   converting tables of data in a source database to components of a target  
3                   application ~~conversion automatically for tables in a database~~, the method  
4                   comprising the ~~following~~ steps:  
5                   obtaining by a computer identification of a) specified tables in said  
6                   source database containing data usable in said target application, b) a target  
7                   location for said target application and c) an application server being used for  
8                   development of said target application at said target location  
9                   ~~invoking a computer application program to read table reading by a~~  
10                  computer definitions of said specified tables from the said source database;  
11                  and  
12                  ~~the computer application program using existing generating by a~~  
13                  computer from said database definitions a plurality of source code files in a  
14                  language of said target application, each said specified table being referenced  
15                  consistently across said plurality of source code files, said plurality of source  
16                  code files including to define object classes and deployment descriptors, said  
17                  specified tables being made accessible to a remote client by said target  
18                  application, said target application being developed using said plurality of  
19                  source code files, and  
20                  ~~the computer application program producing a client-side helper class~~  
21                  ~~to coordinate the name of a database used in a Naming and Directory Service~~  
22                  ~~with client code.~~

FS-00534  
Amendment dated 12/23/2004

09/916,516  
Reply to office action mailed 09/23/2004

02890034aa

1        2. (currently amended) The method of claim 1 further comprising the  
2        ~~following steps of:~~  
3                ~~utilizing providing~~ a user interface to permit ~~the user to interact~~  
4        interaction with the computer application program;  
5                ~~collecting data including in said obtaining step identification of d) an~~  
6        Enterprise JAVA Bean Java Archive (EJB Jar) file;  
7                ~~generating said plurality of source code files in a user-specified~~  
8        directory path; said identified target location, said identified target location  
9        being a user-specified directory path, said application server supporting  
10       development of Enterprise JAVA Beans (EJBs) and supporting a Java  
11       Naming and Directory Interface (JNDI) to assist with said consistent table  
12       references, and said plurality of source code files comprising  
13               ~~generating a Data JAVA file for each specified table, a Home Interface~~  
14       JAVA file for each specified table, a Remote Interface JAVA file for each  
15       specified table, a Bean JAVA file for each specified table, a Primary Key  
16       JAVA file for each specified table, a Persistent JAVA file for each specified  
17       table, an Enterprise JAVA Bean Deployment Descriptor XML file, an  
18       Enterprise JAVA Bean Jar batch command file, a Vendor-Specific  
19       Deployment XML file, and a Vendor-Specific Build batch command file.

1        3. (currently amended) The method of claim 2 ~~for, wherein generating the~~  
2        ~~data collection file providing a user interface further comprises comprising the~~  
3        ~~following steps of:~~  
4                ~~the computer application program querying the database to get names~~  
5        of all tables related to the database, from which each of said specified tables  
6        are selected for the user to encapsulate encapsulation the tables with the an  
7        EJB; and

FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

8 querying the database to acquire information about fields within each  
9 of said specified tables ~~table the user has selected.~~

1 4. (currently amended) The method of claim 2 ~~for, wherein~~ generating the  
2 Data JAVA file ~~comprising the following~~ further comprises the steps of:  
3 generating a file with its filename equal to a table name followed by  
4 the word "Data" and file extension "java";  
5 writing header comments indicating that a JAVA class encapsulates  
6 one row of the ~~selected~~ specified table;  
7 writing a JAVA package definition consistent with the user-specified  
8 directory path;  
9 writing a first JAVA statement to import a java.io.Serializable class;  
10 writing a JAVA class definition with a class name equal to its  
11 filename;  
12 writing a second JAVA statement indicating that the class implements  
13 the Serializable interface;  
14 writing a corresponding attribute definition for each one of a plurality  
15 of fields in the ~~selected~~ specified table;  
16 ensuring that a JAVA data type of each one of a plurality of attributes  
17 is compatible with a database data type of its corresponding field;  
18 ensuring that a name of each one of the plurality of attributes is  
19 identical to the name of its corresponding field;  
20 writing a first JAVA public method to get a value of each one of the  
21 plurality of attributes;  
22 writing a second JAVA public method to set the value of each one of  
23 the plurality of attributes; and  
24 writing a third JAVA public method named "toString" to return a  
25 string representing a Data class.

FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

1        5. (currently amended) The method of claim 2 ~~for, wherein~~ generating the  
2        Home Interface JAVA file ~~comprising the following~~ further comprises the  
3        steps of:  
4                ~~for each one of the plurality of selected tables,~~ generating a file with  
5        filename equal to a table name followed by the word "Home" and file  
6        extension "java";  
7                writing header comments indicating that a JAVA class is the Home  
8        Interface for the Bean encapsulating the ~~selected~~ specified table;  
9                writing a JAVA package definition consistent with the user-specified  
10       directory path;  
11               writing a first JAVA statement to import the following classes:  
12       javax.ejb.EJBHome; javax.ejb.FinderException; javax.ejb.CreateException;  
13       java.rmi.RemoteException; java.util.Collection;  
14               writing a JAVA interface definition with an interface name equal to  
15       the filename;  
16               writing a second JAVA statement indicating that the class extends the  
17       EJBHome interface;  
18               writing a first JAVA method signature named "create" which takes as  
19       parameters the attributes of Data class and returns an object of Remote class;  
20               writing a second JAVA method signature named "create" which takes  
21       as a parameter an object of Data class and returns an object of Remote class;  
22       and  
23               writing a third JAVA method signature named "findByPrimaryKey"  
24       which takes as a parameter an object of PrimaryKey class and returns an  
25       object of Remote class wherein writing the JAVA method signature named  
26       "findByPrimaryKey" has one primary key and the computer application  
27       program makes the parameter its corresponding attribute of Data class.

FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

1        6. (currently amended) The method of claim 2 ~~for, wherein~~ generating the  
2        Remote Interface JAVA file ~~comprising the following~~ further comprises the  
3        steps of:  
4                ~~generating, for each selected table,~~ a file with its filename equal to the  
5        table name followed by the word "Remote" and file extension "java";  
6                writing header comments indicating that this JAVA class is the  
7        Remote interface for the Bean encapsulating the ~~selected~~ specified table;  
8                writing a JAVA package definition consistent with the user-specified  
9        directory path;  
10               writing a first JAVA statement to import the following classes:  
11        javax.ejb.EJBObject; java.rmi.RemoteException;  
12               writing a JAVA interface definition with its interface name equal to  
13        the filename;  
14               writing a second JAVA statement indicating that the class extends the  
15        EJBObject interface;  
16               writing a first JAVA public method signature to get a value of each  
17        attribute and wherein the signature indicates that the method may throw a  
18        Remote Exception; and  
19               writing a second JAVA public method to set the value of each attribute  
20        signature and wherein the signature shall indicate that the method may throw  
21        a Remote Exception.

1        7. (currently amended) The method of claim 2 ~~for, wherein~~ generating the  
2        Bean JAVA file ~~comprising the following~~ further comprises the steps of:  
3                ~~generating, for each selected table,~~ a file with its filename equal to its  
4        table name followed by the word "Bean" and file extension "java";

FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

5           writing header comments indicating that the JAVA class is an EJB  
6           which encapsulates one row of the ~~selected~~ specified table;  
7           writing a JAVA package definition consistent with the user-specified  
8           directory path;  
9           writing a first JAVA statement to import the following classes:  
10          java.rmi.RemoteException; all classes in the javax.ejb package; all classes in  
11          the java.util package;  
12          writing a JAVA class definition with its class name equal to the  
13          filename;  
14          writing a second JAVA statement indicating that the class implements  
15          the EntityBean interface;  
16          writing an attribute definition for one attribute named "theContext" of  
17          type "EntityContext";  
18          writing, for each field in the ~~selected~~ specified table, its corresponding  
19          attribute definition;  
20          ensuring that the JAVA data type of each attribute is compatible with  
21          the database data type of its corresponding field;  
22          ensuring that each attribute is identical to the lower-case name of its  
23          corresponding field;  
24          writing a first JAVA public method to get the value of each attribute;  
25          writing a second JAVA public method to set the value of each  
26          attribute;  
27          writing a third JAVA public method named "setAll" which has an  
28          object of Data class as parameter and can throw a Remote Exception and  
29          wherein each attribute of Bean class is set to the value of its corresponding  
30          Data attribute;  
31          writing a fourth JAVA public method named "getAll" which returns an  
32          object of Data class and can throw a Remote Exception;

FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

33 writing a fifth JAVA public method named "ejbCreate" with attributes  
34 of Data class as parameters and a signature of this method says it returns an  
35 object of type "String" but the body of the method must return "null" and this  
36 method can throw a Create Exception and a Remote Exception;

37 writing a sixth JAVA public method named "ejbPostCreate" with  
38 attributes of Data class as parameters and the signature of this method says it  
39 returns an object of type "String" but the body of the method must return  
40 "null" and this method can throw a Create Exception and a Remote Exception;

41 writing a seventh JAVA public method named "ejbPostCreate" with an  
42 object of Data class as the parameter and the signature of this method says it  
43 returns an object of type "String" but the body of the method must return  
44 "null" and this method can throw a Create Exception and a Remote Exception;

45 writing an eighth JAVA public method named "ejbLoad" which can  
46 throw an EJB Exception or a Remote Exception and the body of this method  
47 may be empty;

48 writing a ninth JAVA public method named "ejbStore" which can  
49 throw an EJB exception or a Remote exception and the body of this method  
50 may be empty;

51 writing a tenth JAVA public method named "ejbActivate" which can  
52 throw an EJB Exception or a Remote Exception and the body of this method  
53 can be empty;

54 writing an eleventh JAVA public method named "ejbPassivate" which  
55 can throw an EJB Exception or a Remote Exception and the body of this  
56 method can be empty;

57 writing a twelfth JAVA public method named "ejbRemove" which can  
58 throw an EJB Exception or a Remote Exception and the body of this method  
59 can be empty;

FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

60                writing a thirteenth JAVA public method named "setEntityContext"  
61                which can throw an EJB exception or a Remote Exception and the parameter  
62                for this method is an object of type EntityContext and is used to set  
63                theContext;  
64                writing a fourteenth JAVA public method named "unsetEntityContext"  
65                which can throw an EJB Exception or a Remote Exception and this method  
66                sets theContext to "null"; and  
67                writing a fifteenth JAVA public method named "toString" to return a  
68                string representing the Bean class.

1                8. (currently amended) The method of claim 2 ~~for, wherein~~ generating the  
2                Primary Key JAVA file ~~comprising the following~~ further comprises the steps  
3                of:  
4                       generating a file ~~for each selected table~~ with at least two primary key  
5                fields such that its filename is made equal to its table name followed by the  
6                word "PrimaryKey" and file extension "java";  
7                       writing header comments indicating that this JAVA class encapsulates  
8                the Primary Key of the ~~selected~~ specified table;  
9                       writing a JAVA package definition consistent with the user-specified  
10                directory path;  
11                       writing a first JAVA statement to import the java.io.Serializable class;  
12                       writing a JAVA class definition with the class name equal to its  
13                filename;  
14                       writing a second JAVA statement indicating that its class implements  
15                the Serializable interface;  
16                       writing a corresponding attribute definition for each primary key field;  
17                       ensuring that the JAVA data type of each attribute is compatible with  
18                the database data type of its corresponding field;



FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

19               ensuring that each attribute is identical to the lower case name of its  
20       corresponding field;  
21               writing a first JAVA public method to get the value of each attribute;  
22               writing a second JAVA public method to set the value of each  
23       attribute;  
24               writing a third JAVA public method named "equals" which returns a  
25       boolean value and the parameter to this method is an object of the Object class  
26       and returns "true" if the parameter is an instance of the Primary Key class and  
27       if the parameter's attributes have values equal to the values of the attributes of  
28       this object; and  
29               writing a fourth JAVA public method named "hashCode" which  
30       returns an integer value and this method forms a String of the attribute values  
31       and returns a hash code of that String.

1       9. (currently amended) The method of claim 2 ~~for, wherein~~ generating the  
2       Persistent JAVA file ~~comprising the following~~ further comprises the steps of:  
3               ~~generating, for each selected table,~~ a file with filename equal to the  
4       table name followed by the word "Persistent" and file extension "java";  
5               writing header comments indicating that this JAVA class encapsulates  
6       one row of the ~~selected~~ specified table and this client side class can be passed  
7       to the server for execution;  
8               writing a JAVA package definition consistent with the user specified  
9       directory path;  
10              writing a first JAVA statement to import the following classes  
11       javax.naming.InitialContext; javax.naming.NamingException;  
12       javax.rmi.PortableRemoteObject; java.sql.Connection; java.io.Serializable; all  
13       classes of the utility package com.lmco.util;

FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

14           writing a JAVA class definition with the class name equal to the  
15       filename;  
16           writing a second JAVA statement indicating that the class extends its  
17       corresponding Data class;  
18           writing a third JAVA statement indicating that the class implements  
19       the Serializable interface and the PersistentObject interface;  
20           writing a first JAVA public method to construct an object of this class  
21       without any parameters;  
22           writing a second JAVA public method to construct an object of this  
23       class with the attributes of the Data class as parameters;  
24           writing a third JAVA public method to construct an object of this class  
25       with an object of the Data class as parameters;  
26           writing a fourth JAVA public method named "create" which returns a  
27       boolean and the body of this method provides a JAVA try-and-catch block to  
28       invoke the "create" method of the Home interface with the attributes of this  
29       class as parameters and if an exception occurs, the result is false or otherwise  
30       the result is true;  
31           writing a fifth JAVA public method named "read" which returns a  
32       boolean and the body of this method provides a JAVA try-and-catch block to  
33       invoke the "findByPrimaryKey" method of the Home interface with the  
34       Primary Key class as parameter and if an exception occurs, the result is false  
35       or otherwise, the result is true and the attributes of this class are set to the  
36       values in the Remote interface returned by the "findByPrimaryKey" method;  
37           writing a sixth JAVA public method named "update" which returns a  
38       boolean and the body of this method provides a JAVA try-and-catch block to  
39       invoke the "findByPrimaryKey" method of the Home interface, with the  
40       primary key class as parameter and if an exception occurs, the result is false

FS-00534  
Amendment dated 12/23/2004

09/916,516  
Reply to office action mailed 09/23/2004

02890034aa

41 or otherwise, the result is true and the attributes of the Remote interface are  
42 set to the values of the attributes in this class;  
43 writing an seventh JAVA public method named "delete" which returns  
44 a boolean and the body of this method provides a JAVA try-and-catch block  
45 to invoke the "remove" method of the Home interface with the Primary Key  
46 class as parameter and if an exception occurs, the result is false or otherwise  
47 the result is true; and  
48 writing a JAVA protected method named "getHome" to return an  
49 object of the Home class and the body of this method providing a JAVA try-  
50 and-catch block to use an object of type InitialContext to look up in the JNDI  
51 a name, which JNDI name is equal to the name of the ~~selected~~ specified table.

1 10. (currently amended) The method of claim 2 ~~for~~ wherein generating the  
2 EJB Deployment Descriptor XML file ~~comprising the following further~~  
3 comprises the steps of:

4 generating a file with filename "ejb-jar.xml" and located in a folder  
5 named META-INF one directory deep within the user-specified target  
6 directory path;

7 writing an XML header statement for documents of type "ejb-jar";

8 writing a first tag <ejb> to begin the document;

9 writing a second tag <description> to begin a description;

10 writing the description of a jar file which includes the package name

11 and the identified EJB jar filename ~~specified by the user~~;

12 writing a third tag </description> to end the description;

13 writing a fourth tag <enterprise-beans> to begin a list of EJB's;

14 writing a fifth tag <entity> to begin a definition of its corresponding  
15 entity EJB;

16 writing a sixth tag <description> to begin the description of the EJB;

FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

17 writing a description of the EJB including the name of its  
18 corresponding ~~selected~~specified table;  
19 writing a seventh tag </description> to end the description;  
20 writing an eighth tag <ejb-name> to begin the name of the EJB;  
21 writing the name of its corresponding ~~selected~~specified table;  
22 writing a ninth tag </ejb-name> to end the name of the EJB;  
23 writing a tenth tag <home> to begin a home of the EJB;  
24 writing a fully-qualified-name of its corresponding Home class;  
25 writing an eleventh tag </home> to end the home of the EJB;  
26 writing a twelfth tag <remote> to begin a remote interface of the EJB;  
27 writing a fully-qualified-name of its corresponding Remote class;  
28 writing a thirteenth tag </remote> to end the remote of the EJB;  
29 writing a fourteenth tag <ejb-class> to begin a bean class of the EJB;  
30 writing the fully-qualified-name of its corresponding Bean class;  
31 writing a fifteenth tag </ejb-class> to end the bean class of the EJB;  
32 specifying container-maintained persistence with the statement  
33 <persistence-type>Container</persistence-type>;  
34 writing a sixteenth tag <prim-key-class> to begin the primary key  
35 class of the EJB;  
36 writing the fully-qualified-name of its corresponding PrimaryKey  
37 class;  
38 writing a seventeenth tag </prim-key-class> to end the primary key  
39 class of the EJB;  
40 specifying the EJB as not re-entrant by writing  
41 <reentrant>False</reentrant>;  
42 beginning each field of the ~~selected~~specified table with <cmp-  
43 field><field-name>;  
44 writing each field of the ~~selected~~specified table;

FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

45                   ending each field of the ~~selected~~specified table with </field  
46                   name></cmp-field>;  
47                   if the ~~selected~~specified table has only one primary key field, writing  
48                   the primary key field as <primkey-field> followed by the field name followed  
49                   by </primkey-field>;  
50                   writing an eighteenth tag </entity> to end a description of its  
51                   corresponding entity EJB for each ~~selected~~specified table;  
52                   writing a nineteenth tag <assembly-descriptor> to begin the assembly  
53                   descriptor;  
54                   writing a twentieth or more tags defining a default security role;  
55                   providing default permission to all methods of all EJB's;  
56                   specifying container-managed transactions for all methods of each  
57                   EJB;  
58                   writing a twenty-first or greater tag <assembly-descriptor> to end  
59                   assembly;  
60                   writing a twenty-second or greater tag </enterprise-beans> to end the  
61                   list of EJB's; and  
62                   writing a twenty-third or greater tag </ejb> to end the document.

1                   11. (currently amended) The method of claim 2 ~~for, wherein~~ generating the  
2                   EJB Jar batch command file ~~comprising the following~~ further comprises the  
3                   steps of:

4                   generating a file with filename set to the ~~user-specified~~identified  
5                   name of the Jar file where the file extension on MS Windows-based systems  
6                   is "bat";  
7                   writing a command to compile each generated Java file;  
8                   writing a command to put the ejb-jar.xml file into one Jar file; and

FS-00534  
Amendment dated 12/23/2004

09/916,516

02890034aa  
Reply to office action mailed 09/23/2004

9                   for each ~~selected~~specified table, writing a command to put all of  
10                   generated JAVA classes into the Jar file.

1                   12. (currently amended) The method of claim 2 ~~for, wherein~~ generating the  
2                   Vendor-Specific Deployment XML file ~~comprising the following~~ further  
3                   comprises the steps of:

4                   generating a file with filename indicating both the vendor and the  
5                   ~~user-specified~~ identified name of the EJB Jar file;

6                   generating one or more XML tags according to vendor specifications  
7                   for a target Application Server and for which the details will vary with the  
8                   Vendor;

9                   generating one or more XML tags to specify a JNDI name for the EJB  
10                  corresponding to each ~~selected~~specified table and the JNDI name is the same  
11                  as its ~~selected~~specified table name; and

12                  generating one or more XML tags to specify mapping of each EJB  
13                  attribute to its field in the ~~selected~~specified database table.

1                   13. (currently amended) The method of claim 2 ~~for, wherein~~ generating the  
2                   Vendor-Specific Build batch command file ~~comprising the following~~ further  
3                   comprises the steps of:

4                   generating a file with its filename indicating both a vendor and the  
5                   word "command" and wherein the file extension on MS Windows-based  
6                   systems is "bat";

7                   generating vendor-specific commands to import and deploy a Jar file  
8                   according to the Vendor-Specific Deployment file.